ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

СОГЛАСОВАНО
Научный руководитель,
профессор департамента
программной инженерии
факультета компьютерных наук
канд. техн. наук

_____ С.М. Авдошин
«___» _____ 2020 г.

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук

_____ В.В. Шилов
«___» _____ 2020 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

ПРОГРАММА КЛАССИФИКАЦИИ КОМПЬЮТЕРНЫХ АТАК ПО НАБОРУ ДАННЫХ ISCX BOTNET

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.02.13-01 81 01-1-ЛУ

Исполнитель
студент группы БПИ193
_____ /Е.А. Гриценко /
«____»_____ 2020 г.

Москва 2020

УТВЕРЖДЕН
RU.17701729.02.13-01 81 01-1-ЛУ

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

ПРОГРАММА КЛАССИФИКАЦИИ КОМПЬЮТЕРНЫХ АТАК ПО НАБОРУ ДАННЫХ ISCX BOTNET

Пояснительная записка

RU.17701729 02.13-01 81 01-1

Листов 28

Москва 2020

ОГЛАВЛЕНИЕ

1. ВВЕДЕНИЕ

1.1. Наименование программы

ПО состоит из двух чатей: библиотеки классов и програмы. Наименование программы — “BotnetDetector”, наименование библиотеки — “LibBtntDtct”. Общее наименование ПО — “BotnetDetector”.

1.2. Документ, на основании которого ведётся разработка

Документом, на основании которого ведётся разработка, является приказ декана факультета компьютерных наук И.В. Аржанцева „Об утверждении тем, руководителей курсовых работ студентов образовательной программы «Программная инженерия» факультета компьютерных наук“ № 2.3-02/1112-04 от 11.12.2019.

2. НАЗНАЧЕНИЕ РАЗРАБОТКИ

2.1. Функциональное назначение

Функциональное назначение разработки соответствует описанному в главе 3.1 Техническго задания.

2.2. Эксплуатационное назначение

Эксплуатационное назначение разработки соответствует описанному в главе 3.1 технического задания.

3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

3.1. Постановка задачи на курсовую работу

- Задание на курсовую работу программы. Документом, на основании которого ведётся разработка, является приказ декана факультета компьютерных наук И.В. Аржанцева „Об утверждении тем, руководителей курсовых работ студентов образовательной программы «Программная инженерия» факультета компьютерных наук“ № 2.3-02/1112-04 от 11.12.2019.
- «Программа классификации компьютерных атак по набору данных ISCX Botnet». Техническое задание.

3.2. Сбор исходных материалов

Требованием к разработке явлась разработка и тестирование данного ПО по набору данных “ISCX Botnet 2014”, закреплённая в специальных требованиях ТЗ. Этот набор данных на момент написания документа доступен по ссылкам в HTML-странице [1] по адресу https://www.unb.ca/cic/datasets/botnet.html (далее — официальном сайте). На той же странице размещалась ссылка для скачивания этого набора http://205.174.165.80/CICDataset/ISCX-Bot-2014/ , проходя по которой производилась процедура регистрации, после чего происходило перенаправление на страницу с listing-ом файлов набора данных. Все эти файлы были скачаны, а именно:

- “ISCX_Botnet-Training.pcap”, размер которого округлённо равен 4.9 гигабайт.
- “ISCX_Botnet-Testing.pcap”, размер которого округлённо равен 2.0 гигабайт.
- “testDset-with iscx.pcap”, размер которого округлённо равен 1.4 гигабайт.
- “listofmaliciousips.docx”, размер которого округлённо равен 4.8 килобайт.

3.3. Предварительный сбор информации об исходных материалах

Официальный сайт [1] сообщает, что этот набор данных был собран на основании следующих данных, а именно: “ISOT dataset”, “ISCX 2012 IDS dataset”, “Botnet traffic generated by the Malware Capture Facility Project” с целью построить обобщённый, реалистичный и репрезентативный набор данных. Указано, что он подразделяется на тестовую и тренировочную часть, для каждой из которых приведён процент вредоносных потоков, а также приведены для каждого набора таблицы, указывающие для каждого типа ботнета процент потоков в этом наборе.

Скачанный же dataset включает в себя три PCAP-дампа трафика и DOCX-файл. Официальная страница сайта [1] упоминает два набора данных: тестовый и тренировочный. При этом, на официальном сайте набора данных не указано, какие из них являются тестовым, какие — тренировочным. Указаны размеры этих наборов данных (5.3 гигабайт для тренировочного и 8.5 гигабайт для тестового), однако ни один из них не соответствует размеру какого-либо из файлов. Таким образом, возникла задача определить какие части дампов являются тестовым, а какие — тренеровочным.

На сайте [1] упомянуто, что полной исследовательской работой, подчёркивающей детали этого набора данных и его подлежащие принципы является “Beigi, Elaheh Biglar, et al. "Towards effective feature selection in machine learning-based botnet detection approaches." Communications and Network Security (CNS), 2014 IEEE Conference on. IEEE, 2014.”. Статья [2] содержит описание и результаты работы, посвящённой определению набора наиболее значимых для выявления botnet-трафика методами машинного обучения характеристик потоков трафика, а также описание набора данных, по которому это определение производилось, полностью идентичное таковому на официальном сайте [1]. Таким образом, указанная на официальном сайте работа не содержит дополнительной информации о наборе данных. Однако непосредственно в статье [2] на странице №282 в уточнении №3 указано, что полная информация об этом наборе данных доступна по адресу “http://iscx.ca/botnet-dataset”. При обращении по этому адресу происходит перенаправление на “http://www.iscx.ca/” [3] не содержащая информации об этом наборе данных, однако же имеющая раздел “Datasets”, при переходе на который происходит перенаправление на “http://www.iscx.ca/datasets/”, содержащее названия доступных в настоящее время наборах данных, в частности, “Botnet data set”, но не содержащая никаких дополнительных ссылок на них [4]. Таким образом, полная информация об этом наборе данных не присутствует по указанной в работе ссылке. Таким образом, обнаружилось отсутствие официальной полной информации о наборе данных.

В результате этого было решено произвести собственный анализ материалов.

3.4. Предварительный анализ имеющихся данных

Для просмотра дампов PCAP-файлов использовалась программа Wireshark. Дампы представляли собой наборы пакетов с их временными отметками. Байты первых и случайно выбираемых пакетов и их представление в кодировке ASCII не несли никакой

очевидной информации о том, были ли они произведены каким-либо из описанных на официальном сайте типов botnet-ов. На сайте [1] также не приведены ссылки на метки к пакетам дампов с этой информацией или способы, по которым такие метки можно проставить. Однако на сайте [1] приведён список вредоносных IP и пар вредоносных IP для IRC. Эта информация идентична таковой в DOCX-файле, за исключением приведённой в нём таблицы вредоносных IP. На основании этого было сделано предположение, что в указанных PCAP-файлах, пакет является вредоносным (порождённым ботнетом) тогда и только тогда, когда IP-адреса этого пакета совпадают с таковыми в приведённом файле.

Для проверки этой гипотезы было решено, как минимум, установить, содержатся ли IP из списка в PCAP-файлах. Ручное выполнение проверки для каждого IP в Wireshark было неудовлитворительно долгим, поэтому проверка была произведена следующим способом.

- Все 35 содержащихся в таблице файла вредоносных IP были построчно записаны в файл “malicious.ips”.
- Из каждого PCAP-файла были выхвачены уникальные IP и создан файл их содержащий. Для этого использовалась команда вида “tcpdump -r 'dataset_part.pcap' ip | cut -d ' ' -f 3 | cut -d '.' -f 1-4 | sort | uniq > dataset_part.ips”. tcpdump — программа для работы с трафиком и его дампами, cut, sort и uniq — часть GNU Coreutils.
- Был написан скрипт “hasips.sh”, генерирующий по двум файлам таблицу с информацией о принадлежности строк первого файла второму. Скрипт принимает первым аргументом файл, и ищет в файле, принимаемом вторым аргументом, строки первого. Если строка первого файла была найдена, в стандартный вывод пишется эта строка и слово “found”, иначе, пишется слово “MISSING”.
- Для каждого из этих файлов, содержащих список уникальных IP, был произведён поиск IP из “malicious.ips” командой “./hasips.sh malicious.ips dataset_part.ips > dataset_part.found”.

Таким образом для каждого из PCAP-файлов была получена таблица, показывавшая, есть ли в нём IP из “listofmaliciousips.docx” или нет. Выяснилось, что в файле “ISCX_Botnet-Training.pcap” содержалось только 10 IP из указанных в DOCX-файле как заражённые. При этом, в Ttraining-файле не содержались IP-адреса, указанные в файле

как порождённые ботнетом Zeus, что противоречит информации на официальном сайте [1], в которой указывалось, что тестовый набор данных содержит следы этого ботнета. В файлах “testDset-with iscx.pcap” и “ISCX_Botnet-Testing.pcap” отсутствовали 4 IP из “listofmaliciousips.docx”, при этом множества имевшихся IP совпадали для этих двух файлов.

Таким образом, гипотеза была опровергнута. Однако на основании этой информации файл “ISCX_Botnet-Training.pcap” был признан тренировочным дэтасетом, файл “ISCX_Botnet-Testing.pcap”, содержащий наибольшее количество найденных IP, — тестовым, поскольку при чтении tcpdump-ом файла “testDset-with iscx.pcap” возникали ошибки, а также потому, что файл “ISCX_Botnet-Testing.pcap” имел больший размер, что более соответствовует публикуемой на сайте информации.

Поскольку

- Во-первых, информация на сайте [1] не соответствовала опубликованным файлам по меньшей мере — об размере этих наборов (если считать размер набора соответствующим размеру файла или суммме размеров файлов), по большем — об IP, присутствующих в этих наборах и отсутствии детальной информации о наборе данных по ссылкам, указанным на официальном сайте и работе на которую он ссылается;
- Во-вторых, PCAP-файлы при ближайшем рассмотрении не содержали никаких очевидных данных, позволявших бы отличить пакеты или сетевые потоки вредоносного трафика от невредоносного (далее — нормального) и тем более выделить отдельные типы ботнетов, описанные на официальном сайте [1].

Было решено отклонить информацию об этом дэтасете, публикуемую на официальном сайте [1] и произвести анализ информации об этом наборе данных в работах, использующих этот дэтасет. Основная цель анализа — получить информацию, позволяющую для каждого потока в наборе определить, является ли он порождённым ботнетом (и какого типа), или же нормальным, необходимую для дальнейшего обучения классификатора.

3.5. Анализ информации о дэтасете в работах его использующих

По результатам из первой страницы запросов “iscx botnet researchgate” в Google и “iscx botnet" в разделе Google книги был произведён поиск информации в следующих работах.

Так “2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT) ‘Predicting Unlabeled Traffic For Intrusion Detection Using Semi-Supervised Machine Learning’” [5] в главе “III Preprocessing Of Dataset” указывает, что нормальный и вредоносный трафик был найден посредством ввода файла “ISCXbotnet.pcap” размером 2 GB (по которому можно понять, что речь идёт о тестовом наборе данных; в рассматриваемой работе о наличии различных наборов данных в составе ISCX Botnet не упоминается) в tcpdump. Однако как именно это было сделано tcpdump-ом не упоминается тоже. tcpdump — программа, обладающая весьма обширным функционалом; определить, как именно он был использован для разметки трафика по тексту работы представляется затруднительным. (tcpdump не содержит классификатора, позволяющего отличать нормальный трафик от вредоносного, в PCAP-файлах очевидной информации о вредоносности трафика не содержится.)

В книге “Cyber Security Cryptography and Machine Learning: First International Conference, CSCML 2017, Beer-Sheva, Israel, June 29-30, 2017, Proceedings”, протоколирующей процесс симпозиума “CSCML 2017, Beer-Sheva, Israel, June 29-30, 2017, Proceedings” на странице 261 с отрывком статьи “Learning Representations for Log Data in Cybersecurit” [6], её авторами было описано произведение разметки потоков набора данных на основании публикуемых IP. Однако в предылущих главах этой пояснительной записки показано, что в силу выявленной недостоверности и недостаточности информации об этом наборе данных, точность таковой разметки сомнительна. Поэтому производится поиск более надёжного метода разметки.

В “Botnet analysis using ensemble classifier.” [7] не пояснеется процесс разметки. Какой из наборов данных использовался также не поясняется.

В “Hybrid Botnet Detection Based on Host and Network Analysis” [8] процесс разметки также не поясняется. Судя по размеру файла, использовался тренировочный дэтасет.

В “Synthetic Minority Oversampling Technique for Optimizing Classification Tasks in Botnet and Intrusion-Detection-System Datasets” [9] не упоминается о том, что ISCX Botnet 2014 состоит из тестовой и тренировочной частей, и трёх дампов; однако упаминаются типы ботнетов, не встречающиеся в тренировочном наборе данных, из чего можно сделать вывод, что либо использовался тренировочный дэтасет, либо некая конкатенация дампов

дэтасета. При этом они упоминают размер дэтасета как 5.3 GB, поэтому можно было бы сделать вывод, что дэтасет, который они используют — тренировочный, однако он, согласно информации на официальном сайте, не содержит следы большинства упоминаемых ими ботнетов. Процесс разметки не расписан в той мере, которая позволяет его реплицировать или даже приблизительно понять то, как он производился.

В “Botnet Detection Using On-line Clustering with Pursuit Reinforcement Competitive Learning” [10] упоминается о тренировочном дэтасете как о дэтасете размером 5 GB и о тестовом детасэте как о дэтасете размером 2 GB, что соотносится с выявленным; хотя и у этих авторов нет упоминания о наличии PCAP-дампа размером 1.3 GB. Процесс разметки также не описан в достаточной для репликации форме. Статья по большей мере повторяет статью авторов дэтасета на конференции IEEE.

Из шести рассмотренных работ, только в одной приводится метод разметки трафика. Работы содержат недостаточные или противоречивые сведения о составе дэтасета и его размере. Поэтому, было сделано предположение, что дальнейший поиск информации о дэтасете в иных научных работах также крайне неэффективен, на основании чего этот поиск был прекращён.

Как следствие, было решено произвести разметку следующим образом:

- Сопоставить попакетно части дампов дэтасета “ISCX Botnet 2014” частям дэтасетов, из которых он был скомпилирован.
- Для каждой из частей этих дэтасетов произвести разметку пакетов трафика.
- На основании попакетного сопоставления приравнять метки пактов частей, из которых состоят дэтасеты “ISCX Botnet 2014”, соответствующим пакетам трафика дэтасетов “ISCX Botnet 2014”.

3.6. Поиск исходников дэтасета “ISCX Botnet 2014”, их сопоставление и разметка

Согласно информации на официальном сайте [1], дэтасет был составлен по данным из трёх источников: “ISOT dataset”, “ISCX 2012 IDS dataset”, “Botnet traffic generated by the Malware Capture Facility Project”, а в случае тестового — дополнен ещё и неким иным трафиком. Разметку было решено начать с тренеровочного дэтасета.

Дэтасет “ISCX 2012 IDS” описан на том же сайте на странице “https://www.unb.ca/cic/datasets/ids.html” [11], ссылка, предлагающая его скачивание, находится внизу страницы (“http://205.174.165.80/CICDataset/ISCX-IDS-2012/”). Переход

на неё вызывает пренаправление на форму регистрации, после прохождения которой, происходит перенаправление на станицу с listing-ом файлов набора данных, которые были скачаны. Это файлы:

- “labeled_flows_xml.zip”, с округлённым размером 293.4 MB
- “testbed-11jun.pcap”, с округлённым размером 16.1 GB
- “testbed-12jun.pcap”, с округлённым размером 4.2 GB
- “testbed-13jun.pcap”, с округлённым размером 4.0 GB
- “testbed-14jun.pcap”, с округлённым размером 6.9 GB
- “testbed-15jun.pcap”, с округлённым размером 23.4 GB
- “testbed-16jun.pcap”, с округлённым размером 12.4 GB

Файл “labeled_flows_xml.zip” содержал XML-файлы, названия которых соответствовали названиям PCAP-файлов, с информацией о потоках, файлы с расширениями “xsd”, названия которых без расширения были идентичны названиям XML-файлов без расширения, а также “readme.txt” с описанием: какая сетевая активность производилась в дни захвата трафика, длительности следов трафика, извлечённых feature потоков.

Поиск не выявил программы, позволяющие создавать нумерованное сопоставление пакетов одного дампа пакетам другого. Поэтому для такового поиска была разработана программа “TestPcapInterconnectivity.exe”, ищущая пакеты различных дампов (далее — «стог») в указанам дампе (далее — «ига»), и генерирующая CSV-таблицу сопоставления со строками формата «номер_пакета_в_игле, путь_к_дампу_стога_которому_принадлежит_пакет, номер_пакета_в_дампе_стога». Нумерация, начинающаяся с нуля, осуществляется по порядку, в котором происходит чтение пакетов библиотекой SharpPcap. Программа полностью загружает иглу в память, загрузить стог полностью не представлялось возможным в силу его размеров значительно превышающих размеры памяти доступных технических средств. При этом в процессе разработки оказалось, что чтение пакетов по-одному значительно замедляет работу программы, поэтому было организовано чтение чанками по 2000000 (два миллиона) пакетов. Сопоставление было решено производить на основании побайтового равенства пакетов. В процессе разработки и впоследствие было использовано несколько реализаций алгоритма поиска:

- Для каждого дампа стога производилось почанковое чтение, после чтения чанк сортировался встроенными средствами .NET по написанному алгоритму сравнения массивов байт, многопоточно для каждого пакета иголки производился бинарный поиск этого пакета в чанке, и если пакету был ранее сопоставлен другой пакет, то увеличивалось число коллизий, иначе пакету иголки присуждалось соответствие найденному пакету текущего дампа, затем чанк выгружался из памяти.

- Пакеты иголки сортировались (тем же образом), затем для каждого дампа стога производилось почанковое чтение, после чтения чанк сортировался, и происходил многопоточный поиск пакетов чанка в игле методом двух указателей, если пакету иглы было уже ранее присвоено некое сопоставление, то увеличивалось число коллизий.

В результате выполнения программы для PCAP-дампов IDS-dataset-a (стог) и дампа тренировочного набора ISCX Botnet (игла) было обнаружено, что все пакеты иглы, начиная с номера 3414290, побайтово равны каким-то из пакетов стога, и никакие другие пакеты иглы таким свойством не обладают. Более того, в игле обнаружились только пакеты из файлов “testbed-11jun.pcap” и “testbed-12jun.pcap”. После этого поиск был выполнен только для “testbed-11jun.pcap”, затем только для “testbed-12jun.pcap”, в ходе которого выяснилось, что пакеты иглы с 3414290-ого по последний 9 288 269-ый являются какими-то пакетами дампа “testbed-12jun.pcap”. При этом при просмотре сгенерированного файла и кусков дампов в Wireshark было выявлено, что скорее всего эта часть иглы является цельным куском дампа “testbed-12jun.pcap”, впоследствие подтвердившееся.

Далее сопоставление было решено произвести для ISOT-дэтасета. По первым строкам запроса “isot dataset” был найдена страница, посвящённая различным ISOT-дэтасетам: https://www.uvic.ca/engineering/ece/isot/datasets [12]. Оказалось, что существует несколько различных дэтасетов, имеющие в названии ISOT, из них два относятся к ботнетам: “ISOT Botnet Dataset”, “ISOT HTTP Botnet Dataset”. Было сделано решение проверить первый дэтасет, оказавшееся верным. Для скачивания была использована сслыка http://www.isot.ece.uvic.ca/dataset/isot_botnet.php , приведённая на найденной странице. Был скачан архив “ISOT_Botnet_DataSet_2010.tar.gz”, файлами которого являются:

- “ISOT_Botnet_DataSet_2010.pcap”, с округлённым размером 10.6 GB;

- “ISOT Dataset Overview-v0.5.pdf”, с округлённым размером 128.9 KB.

PDF-файл содержал подробное описание этого дэтасета, в частности, описание меток заражённого трафика. Также в описании была приведена таблица с IP заражённых и не заражённых компьютеров, в которой для заражённых указывался тип ботнета и метка заражённого трафика: пакеты заражённого трафика помечались определёнными MAC-адресами. При этом для SMTP-спама, генерируемого ботнетами, наличествовало деление на спам, генерируемый ботнетом Storm, и спам генерируемым ботнетом Waledac, в то время как типы ботнетов на официальном сайте дэтасета “ISCX Botnet 2014”, судя по всему, объеденяли их в одну категорию “SMTP Spam”. Помимо этого в ISOT содержался UDP-трафик, генерируемый ботнетом Storm, класс которого не присутствовал в тренировочном наборе данных.

Был произведён поиск пакетов иглы в PCAP-файле ISOT-дэтасета вмете с файлом “testbed-12jun.pcap” ISCX IDS дэтасета (стоге) описанной ранее программой. В результате выяснилось, что пакеты ISOT-дэтасета занимают пакеты иглы с 1216329-ого по 3414289-ый включительно. При этом из ISOT-дэтасета изымались только некоторые последовательности пакетов, поскольку размер иглы был приблизительно в два раза его меньше.

Далее был произведён поиск в интернете ссылок для скачивания трафика, сгенерированного “Malware Capture Facility Project” (далее MCFP), и его описания. Эта задача оказалась весьма нетривиальной, поскольку на сайте, найденном по первым строкам результата поискового запроса, являюшегося сайтом общего проекта, к которому принадлежит искомый: https://www.stratosphereips.org/datasets-malware [13] оказалось, что MCFP включает в себя более чем 300 (три сотни) различных наборов данных ботнет-трафика, на официальном сайте же не было указано, какой именно из этих трёх сотен наборов был использован. В кратком описании, приводившемся ко многим наборам в списке на найденном сайте, не были обнаружены типы ботнетов, указанные для “ISCX Botnet 2014” [1]. Официальный сайт при упоминании трафика, сгенерированного MCFP, ссылался (в сноске №8) на ‘S. Garcia, “Malware capture facility project, retrieved July 03, 2013. university,”’ и приводил ссылку на сайт https://mcfp.agents.fel.cvut.cz/ , сервер которого не был найден и не находится на момент написания этого документа. По запросу “S. Garcia Malware capture facility project, retrieved July 03, 2013. university” был найден

сайт http://agents.fel.cvut.cz/malware-capture-facility , представлявший себя как официальный сайт MCFP. При этом на сайте [14] в изложении правил цитирования упоминался полный дэтасет (возможно, все 300+ «более мелких» дэтасетов), а также наличие специфичных дэтасетов (возможно, какой-то конкретный из сотен). Однако, для цитирования обоих упоминалась дата “February 03, 2013”, дате “July 03, 2013” не соответствовавшая, из чего было сделано предположение, что ISCX были взяты конкретные наборы данных из полного дэтасета, соответствовавшие указанной ими дате. Перейдя по ссылке на директорию с набором данных “CTU-Malware-Capture-Botnet-1” (ни дата которого, ни какие-либо его пакеты не соответствовали побайтово пакетам иглы), а далее — в родительскую директорию с полным listing-ом директорий публикуемых дэтатасетов, для 25 первых дэтасетов (их даты обычно возрастали с возрастанием номера) была сделана проверка readme-файлов (если они был доступны) на соответствие указанной ISCX дате. Таковых наборов данных не было. При этом для набора данных “CTU-13” указанное в readme описание не загружалась, а его номер совпадал с номером года, указанного ISCX, поэтому на этот набор данных было решено обратить более пристальное внимание. На сайте «родительского» для MCFP проекта был найден раздел CTU-13 [15], в котором описывался набор данных с типами ботнетов, соответствовавший упоминаемым ISCX. Поэтому было сделано предположение, что трафик этого дэтасета входит в состав “ISCX Botnet 2014”. Этот набор данных был скачан в виде архива весом 1.9 GB по ссылке https://mcfp.felk.cvut.cz/publicDatasets/CTU-13-Dataset/CTU-13-Dataset.tar.bz2 , приведённой на той же странице.

Архив содержал тринадцать пронумерованных директорий с дампами, потоками и метками к ним, а также описанием того, как производился захват. В частности, в readme-файлах каждой директории упоминалось, что дампы содержат только вредоносный трафик, трафик невредоносный не публиковался в силу конфиденциальности. При этом каждый из составляющих CTU-13 наборов данных датирован 11-ым годом, что не соответствовало публикуемой в ссылке ISCX дате захвата. Тем не менее, в силу того, что этот дэтасет был единственным найденным из MCFP, со следами типов ботнетов таких же как и те, что были указаны иглы, было решено произвести побайтовый поиск в нём пакетов иглы той же программой.

Результаты поиска показали, что некоторые пакеты иглы действительно побайтово равны некоторым пакетам этого набора данных. Поскольку это был неполный (содержащий только вредоносный трафик) CTU-13 дэтасет была произведена такая же проверка для расширенного CTU-13 дэтасета MCFP [16], содержашего обрезанные заголовки пакетов всего трафика, однако для него программа вовсе не нашла никаких соответствий.

При просмотре и ручном сопоставлении иглы с дампами CTU-13, пакеты которых обнаружились в игле, было установлено, что пакеты иглы, за исключением как правило контрольных сум (видимо, их изменение произошло в результате replaying-а трафика), побайтово равны всем пакетам некоторых дампов CTU-13, и более того, временные метки полностью совпадают.

Для проверки этой гипотезы программа была переписана таким образом, чтобы поиск пакетов производился по временным меткам пакетов. Принималось, что временные метки пакетов иглы и дампов стога возрастают, после чего производилось сопоставление методом двух указателей. В результате выполнения программы как для отдельных дампов из CTU-13, так и для нескольких, было установлено, что пакеты иглы с 0-ого по 1216329-ый являются последовательной конкатенацией дампов 1-ой, 3-ей, 5-ой и 12-ой директорий. Таким образом, все дампы, из которых состоит тренировочный набор данных были найдены. Для ISOT-дэтасета произвести подобное не оказалось возможным, поскольку его пакеты датированы раньше, чем CTU-13, и также не являются строго возрастающими. По подобной причине не было возможности тем же алгоритмом произвести сопоставление по времени для testbed-12jun.pcap. Разработка алгоритма, способного работать с непоследовательными временными метками, отняла бы много времени, вследствие чего было решено оставить имеющуюся программу как есть.

Было решено несколько улучшить последовательность сопоставления, расчитанную для ISOT-дэтасета. Для этого для части иглы (вырезание частей дампа производилось программой editcap), соответствовавшей ISOT-дэтасету была написана программа, которая пока может, пытается производить на основании сгенерированного предыдущей программой сопоставления, проводить его последовательно. Результатом стала таблица сопоставления пакетов иглы из ISOT-дэтасета пакетам ISOT-дэтасета.

Для части иглы, содержащей “testbed-12jun.pcap”, был выполнен поиск пакетов по времени в “testbed-12jun.pcap”. Результатом стала таблица сопоставления пакетов для этой части иглы. При этом оказалось, что “testbed-12jub.pcap” целиком входит в тестовый дэтасет.

Для части иглы, содержащей пакеты дампов CTU-13 также была получена такая таблица.

Была написана программа, позволяющая конкатенировать эти таблицы, в результате которой был получена итоговая таблица сопоставления для файла “ISCX_Botnet-Testing.pcap”.

Была попытка произвести подобные операции и для тренировочной части набора, однако это оказалось слишком трудозатратным, во-первых, в силу того, что пакеты идут по времени менее последовательно, во-вторых, были найдены области, не соответствующие никаким из трёх наборов данных, которые пришлось бы обрабатывать отдельно, что также было нецелесообразно длительным процессом. Поэтому разметку пакетов тестового набора было решено делать на основании публикуемых IP, хоть она и не является надёжной.

Для ISOT-дэтасета была написана программа, генрировавшая таблицу со строками вида «номер_пакета_в_тестовом_дэтасете, метка_типа_ботнета». Назовём такую таблицу таблицей меток. Генерация происходила на основании указаных в его описании вредоносных MAC-адресов.

Поскольку в описании набора данных ISCX IDS говорилось, что в день захвата “testbed-12jun.pcap” не производилась вредоносная активность, все его пакеты принимались за невредоносные. Поскольку в описаниях к дампам трафика CTU-13 говорилось, что весь трафик этих дампов является вредоносным и то, что вредоносным считался весь трафик с определённых IP, была написана программа, которой по файлам с парами «ip, метка» были построены таблицы меток для дампов дэтасета CTU-13. Она же использовалась для построения таблицы меток для “testbed-12jun.pcap” (если IP пакета не было в файле с парами «ip, метка», то пакет помечался как “normal”).

Была составлена программа, позволяющая по таблице сопоставления и таблицам меток для каждого из упомянутых в ней дампов генерировать таблицу для файла, для

которого производилось сопоставление. В результате работы этой программы была получена таблица меток для тренировочного дэтасета.

3.7. Определение термина «сетевой поток» (“flow”) и способа их выявления

Анализ доступной в интернете информации выявил, что понятие сетевого потока “network flow” является весьма нечётким.

Например, документ RFC 3917 упоминает о существовании различных определений термина “flow”, использующегося интернет сообществом [17]. Может производиться деление потоков на однонаправленные и двунаправленные. Так, в readme-файлах к дампам CTU-дэтасета производится чёткое деление потоков на однонаправленные (unidirectional) и двунаправленные (bidirectional). Стандарт RFC 5103 [18] определяет однонаправленный поток как состоящий из пакетов, отправленных от одной конечной точке к другой конечной точке, двунаправленные — как пакеты, состоящие из пакетов двух однонаправленных потоков таких, что их не-направленные ключи совпадают, а их направленные ключи являются инверсией друг друга.

При этом таковое деление производится не всегда, и “flow” трактуется либо как однонаправленный поток, либо как двунаправленный. Так Википедией [19] без ссылок на стандарты, утверждается, что протокол NetFlow версии 5 компании Cisco использует именно однонаправленные потоки. На официальном сайте Cisco эту информацию, как и вообще какую-либо точную информацию о том, что считается ключами, определяющими принадлежность пакета к потоку, найти не удалось. Документ RFC 3954, описывающий 9-ую версию этого протокола, прямо говорит о том, что детали процесса вявления потока выходят за рамки документа [20].

С другой стороны, авторами ISCX Botnet-дэтасета, судя, в частности, по использованию такого свойства потоков, как IOPR — отношение числа входящих пакетов к числу исходящих [2], осмысленного только в случае, если поток двунаправленный, под “flow” понимается именно biderectional flow, и деление на однонаправленные и двунаправленные потоки не производится.

Также, например, могут быть различия и в обработке флагов протоколов. Например, ранее упомянутый RFC 3954 говорит о том, что TCP-поток может считаться просроченым при детекции флага FIN или флага RST [21]. Однако, например, программа ISCXFlowMeter, предназначенная для генерации и анализа потоков Ehernet-трафика и

использованная в наборах данных ISCXVPN2016 и ISCXTor2016 [22], судя по коду метода addPacket класса FlowGenerator.java [23], считает поток завершённым при первом же наличии флага FIN, игнорируя четверное рукопожатие протокола TCP, и совершенно не обрабатывает флаг RET.

В работе авторов набора данных “ISCX Botnet 2014” производилось извлечение свойств потоков в течение временного окна в 60 секунд, дававшее наибольшую точность [2]. Поскольку чёткого описания процесса извлечения авторами приведено не было, было решено, что по истечении минуты все потоки считаются завершёнными.

На основании проанализированной информации, в рамках разработки данного ПО, было использовано следующее понятие «сетевого потока». Под «сетевым потоком» в рамках разработки данного ПО понимается упорядоченная по времени последовательность Ethernet-пакетов с IP-payload-ом в рамках определённого временного интервала, по истечении которого поток считается завершённым, обладающая следующими характеристиками:

- IP адрес и порт получателя и отправителя соответствует таковым получателя и отправителя или [наоборот] отправителя и получателя у первого пакета. В случае, если протокол третьего уровня не является TCP или UDP, порты считаются равными 0.

- Протокол четвёртого уровня (по OSI) совпадает с таковым у первого пакета.

С одной стороны, такая модель потока совершенно не учитывает особенности протоколов четвёртого уровня и финализации их соединений, вследствие чего множество потоков приложения, часто разрывающего соединение и восстанавливающего их по тому же порту, будет считаться одним, с другой, она проста в реализации. К тому же, поскольку флаги завершения соединения игнорируются, это возможно позволит точнее выявить ботнет трафик, в случаях когда бот часто разрывает соединения и восстанавливает их по тем же портам.

3.8. Определение свойств захватываемой информации о потоке и алгоритма классификации

Авторами набора данных “ISCX Botnet 2014” для классификации использовалось дерево решений, обучаемое алгоритмом C4.5 с последующим pruning-ом [2]. Также ими был произведён поиск различных свойств потоков, использование которых для обучения дерева даёт наибольшую точность классификации; макисмальная точность, достигнутая

ими на тренировочном наборе данных составила 99%, на тестовом — 75%, для этого использовались такие свойства потоков, как длительность, средняя длина payload-а, отношение числа входящих к пакетов к числу исходящих и средняя скорость в битах в секунду.

В качестве классифиактора и алгоритма его обучения было решено взять такие же. В качестве характеристик потоков было решено взять похожие:

- Длительность, определяемая как разница между временем последнего и первого пакета в потоке; елси она была равна 0 (например, если в потоке только один пакет), то её значение считалось равным 0.01 секунды.

- Средняя длина payload-а. Рассчитываемая как сумма всех длин payload-ов пакетов потока делённая на их количество. В случае, если протоколы 4-ого уровня были UDP или TCP, за длину payload-а пакета бралась длина их payload-а. Иначе, бралась длина payload-а IP-пакета.

- Скорость в октетах в секунду. Рассчитываемая как сумма октет во всех Ethernet-фрэймах потока делённая на длительность.

- Отношение числа исходщих пакетов к общему их числу. Пакет потока считается исходящим, если его получатель и отправитель (их IP и порт) совпадают с получателем и отправителем первого пакета в потоке.

На основании этого списка свойств, было решено, что ПО должно выделять из трафика или его дампа следующую информацию о потоке: дату и время первого пакета, дату и время последнего пакета, ip-адрес отправителя, порт отправителя, ip-адрес получателя, порт получателя, номер ip-протокола, число пакетов, число исходящих пакетов, число октет, число октет в payload-е пакетов. Производить классификацию было решено по этой информации на основании указанных выше характеристик, из этой информации получаемых.

Для выявления из тренировочного дэтасета информации о его потоках и добавлении к ним меток, согласно таблице меток пакетов, была создана программа, создающая по PCAP-файлу и таблице меток таблицу с информацией о потоках и метках к ним. Формат этой таблицы определил форматы входных и выходных данных разрабатываемого ПО. Подробная информация об этих форматах находится в главе 4.1.2 технического задания.

Для выявления из тестового дэтасета информации о его потоках и добавлении меток к ним была составлена программа, которая на основании типов ботнетов и их IP из “listofmaliciousips.docx”, составляла таблицу с информацией о потоках и метках к ним.

Таким образом завершилась предобработка тренировочного и тестового наборов данных и были получены окончательный тренировочный и тестовый наборы данных, непосредственно использовавшиеся для обучения дерева решений, и представлявшие собой таблицы с информацией о потоках и метках к ним. Итоговый тренировочный набор содержит информацию о 290574 потоках, из которых 197854 (68%) не помечены как вредоносные. Итоговый тестовый набор содержит информацию о 331303 потоках, из которых 176808 (53%) не помечены как вредоносные.

3.9. Создание программы, обучающей классификатор и выбор библиотеки машинного обучения

В качестве библиотеки для обучения была использована библиотека .NET Accord.Framework, позволявшая обучать дерево решений алгоритмом C4.5, а также позволявшая представлять обученное дерево в виде C# кода. Представление дерева решений в виде нативного C# когда позволяет увеличить производительность, поскольку фактически классификация в таком случае сводится исключительно к проведению операций сравнения и jump-операций, количеством соответствующих глубине дерева. Также это позволяет избежать зависимости от библиотеки машинного обучения в итоговом ПО. Помимо этого, Accord.Framework позволяет задавать максимальную глубину дерева, что даёт удобный инструмент контроля времени обучения и возможность избежать переобученние классификатора, а следовательно, и необходиости в pruning-е. На основании этих факторов для обучения был выбран Accord.Framework.

На основании этой библиотеки, модифицированной таким образом, чтобы она выводила прогресс (текущую глубину рекурсии, дерево строилось рекурсивно), была создана программа

- По заданной таблице с потоками и метками строящая дерево решений по 4-м переменным (выбранным характеристикам потока) и возвращающее индекс класса потока. Класс потока определялся как “malicious”, если метка не “normal”. Максимальная глубина дерева определяется заданной в коде константой. Программа сохраняет дерево

сериализатором библиотеки после его обучения и выводит отношение ошибочно классифицированных потоков к общему их числу.

- По заданной таблице с потоками и метками выполняющая указанный файл дерева в формате сериализатора библиотеки, и вычисляющая процент ложно-положительно, ложно-отрицательно, истинно-положительно и истинно-отрицательно классифицированных потоков, а также число и процент потоков в определённых классификатором классах и классах, прочитаных в таблице. Классфикация производится бинарная: “malicious” или “normal”. Поток считается “malicious”, если его метка не равна “normal”. Результат считается положительным, если поток был классифицирован как “malicious”.

- Конвертировало указанное дерево в формате сериализатора библиотеки в C#-коде.

3.10. Результаты обучения классификатора

Ниже приведена статистика точности обученного программой классификатора для различной глубины дерева:

Глубина дерева.	Точность на тренировочном наборе.	Точность на тестовом наборе.
12	95.834%	61.085%
14	96.854%	61.962%
16	97.327%	60.752%
18	97.682%	60.301%

Для встраивания в разрабатываемое ПО был выбран классификатор с глубиной дерева 16. Более полные результаты его тестирования на тестовом наборе данных приведены в таблице ниже. Как уже упоминалось, итоговый тестовый набор содержит информацию о 331303 потоках, из которых 176808 (53%) не помечены как вредоносные.

TP	TN	FP	FN	Классифицированы как “normal”	Классифицированы как “botnet”
20.292%	40.459%	12.909%	26.341%	66.799%	33.201%

Классификатор был сконвертирован программой в C#-код и был дополнен кодом, необходимым для его встраивания в систему классов разрабатываемого ПО.

3.11. Тестирование классификатора на иных наборах данных

После окончания разработки ПО было решено протестировать встроенный классификатор (с глубиной дерева 16) на иных наборах данных. При помощи ПО была захвачена информация о потоках из трафика живого сетевого устройства, а именно: торрент-трафик и трафик, захваченный во время просмотра различных веб-сайтов, изображений и видео на них, в основном — видео с youtube.com. Таким образом, весь захваченный трафик не является вредоносным. Ниже представлена таблица с полученными результатами.

Тип трафика.	Процент потоков, определённый как ботнет.
Торрент.	43.824%
Web-browsing	12.005%

При этом весь упомянутый в таблице трафик не является вредоносным или порождённым ботнетом, таким образом процент ложно-положительных результатов может доходить как минимум до 43% на определённых типах невредоносного трафика.

3.12. Описание и обоснование выбора требований к техническим и програмным средствам

Для работы с пакетами была выбрана .NET — библиотека SharpPcap, предоставляющая интерфейс к библиотекам npcap или libpcap, и таким образом являющаяяся кроссплатформенной для Windows и основанных на GNU/Linux систем. ПО использует эту библиотеку, поэтому в состав требований включено требование обеспечить её корректную работу. ПО основано на .NET Framework 4.7, поэтому его

реализация также должна быть установлена и работоспособна, это обстоятельство также отражено в требованиях к программе в ТЗ [25] и условиях выполнения программы, описанных в руководстве оператору [26]. Поскольку программа использует библиотеки классов, реализация .NET должна их корректно загружать, это обстоятельство также описано в условиях выполнения программы в руководстве оператору.

Жёсткие требования к архитектуре процессора было решено не включать, поскольку в случае корректной работы реализации .NET Framework 4.7 и библиотек-зависимостей, работоспособность разработанного ПО должна быть в целом инвариантна архитектуре процессора. Работоспособность ПО была проверена на процессорах микроархитектур IvyBridge и ARMv8-A (эмулированна с использованиям ПО QEMU), поэтому x86-64 и aarch64 включены как примеры допустимых архитектур в техническое задание.

Оперативная память, занимаемая программой при низкой нагрузке (прослушивание живого трафика, в очереди почти нет пакетов), колебалась в диапазоне от 128 MB до 256 MB, поэтому 256 MB было решено включить как минимальную оперативную память, которую программе должна быть способна выделить ОС.

Поскольку необходимая память и вычислительные мощности, необходимые для работы программы зависят от интенсивности трафика (числа пакетов в секунду, количества потоков и октет), это обстоятельство также было решено включить в техническое задание и условия выполнения программы.

4. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

4.1. Требуемые вычислительные мощности

Детальное исследование требуемых вычислительных мощностей в зависимости от интенсивности трафика не производилось. Однако на ЭВМ с 16 GB оперативной памяти и двухядерным процессором Intel(R) Core(TM) i7-3520M и тактовой частотой ядра при максимальной загрузке всех ядер 3.4 GHz.

- При анализе торрент трафика со скоростью около мегабайта в секунду, одно из виртуальных (HyperThreading) ядер, было загружено ПО на 2.5%,
- При анлизе трафика, порождённого скачиванием программой youtube-dl видеоданных, размером 480.38 MB на средней скорости 17 MB в секунду, ПО загружало одно из ядер на 60%. При этом число пакетов в очереди не превышало 5000 и периодически вовсе обнулялось.
- При анализе таблицы с информацией о 317977 сетевых потоках, таблица с информацией о них и их классах была сохранена и сгенерирована за 7.576 секунды.

Эта информация может быть использована для определения необходимых вычислительных мощностей в зависимости от интенсивности трафика.

4.2. Некоторые возможные пути использования

Программа может использоваться для тестирования классификаторов, использующих то же определение потока (“flow”) и производящих классификацию на основании информации о потоке, захват которой предусмотрен ПО.

Вывод программы может использоваться для генерации данных о сетевых потоках как для последующей классификации, так и для использования другими программами. Поскольку выходные файлы-таблицы с информацией о потоках (или потоках и их классах) соответствует формату CSV (comma separated values), возможна их обработка, просмотр и редактирование в более наглядной табличной форме, например, программами Microsoft Excel (при соответствующих настройках) и LibreOffice Calc.

Если командная оболочка, в которой запускается ПО, поддерживает перенаправление потоков вывода, то вывод программы во время её выполнения может быть использован другими программами и скриптами, например, для вывода уведомления (alarm) в случае, если зафиксировано аномальное количество классифицированного как “botnet” трафика.

4.3. Расчёт экономических показателей

Расчёт экономических показателей не производился.

5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Botnet 2014 | Datasets | Research | Canadian Institute for Cybersecurity | UNB // URL: https://www.unb.ca/cic/datasets/botnet.html (дата обращения: 13.05.2020).

2. Beigi, Elaheh Biglar, et al. Towards effective feature selection in machine learning-based botnet detection approaches. // Communications and Network Security (CNS), 2014 IEEE Conference on. IEEE, 2014.

3. Information Centre of Excellence for Tech Innovation // URL: http://www.iscx.ca (дата обращения: 13.05.2020)

4. Datasets — ISCX // URL: http://www.iscx.ca/datasets (дата обращения: 13.05.2020)

5. Chidananda Murthy P, A S Manjunatha, Anku Jaiswal, Madhu B R Predicting Unlabeled Traffic For Intrusion Detection Using Semi-Supervised Machine Learning // 2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT)

6. Ignacio Arnaldo, Alfredo Cuesta-Infante, Ankit Arun, Mei Lam, Costas Bassias, Kalyan Veeramachaneni Learning Representations for Log Data in Cybersecurit // Cyber Security Cryptography and Machine Learning: First International Conference, CSCML 2017, Beer-Sheva, Israel, June 29-30, 2017, Proceedings

7. Anchit Bijalwan a, Nanak Chand b , Emmanuel Shubhakar Pilli c , C. Rama Krishna b Botnet analysis using ensemble classifier.

8. Saoucene Mahfoudh, Sultan Almutairi, and Jalal S. Alowibdi Hybrid Botnet Detection Based on Host and Network Analysis

9. David Gonzalez-Cuautle, Aldo Hernandez-Suarez, Gabriel Sanchez-Perez, Linda Karina Toscano-Medina, Jose Portillo-Portillo, Jesus Olivares-Mercado, Hector Manuel Perez-Meana, and Ana Lucila Sandoval-Orozco Synthetic Minority Oversampling Technique for Optimizing Classification Tasks in Botnet and Intrusion-Detection-System Datasets.

10. Yesta Medya Mahardhika, Amang Sudarsono, Aliridho Barakbah Botnet Detection Using On-line Clustering with Pursuit Reinforcement Competitive Learning.

11. IDS 2012 | Datasets | Research | Canadian Institute for Cybersecurity | UNB // URL: https://www.unb.ca/cic/datasets/ids.html (дата обращения: 13.05.2020)

12. Datasets University of Victoria // URL: https://www.uvic.ca/engineering/ece/isot/ datasets (дата обращения: 13.05.2020)

13. Malware Capture Facility Project — Stratosphere IPS // URL: https://www.stratosphereips.org/datasets-malware (дата обращения: 13.05.2020)

14. Malware Capture Facility | Agent Technology Center // URL: http://agents.fel.cvut.cz/malware-capture-facility (дата обращения: 13.05.2020)

15. The CTU-13 Dataset. A Labeled Dataset with Botnet, Normal and Background traffic. — Stratosphere IPS // URL: https://www.stratosphereips.org/datasets-ctu13 (дата обращения: 13.05.2020)

16. Index of /publicDatasets/CTU-13-Extended-Dataset // URL: https://mcfp.felk.cvut.cz/publicDatasets/CTU-13-Extended-Dataset/ (дата обращения: 13.05.2020)

17. RFC 3917 Requirements for IP Flow Information Export (IPFIX) секция 2 “Terminology” // URL: https://tools.ietf.org/html/rfc3917 (дата обращения: 13.05.2020)

18. RFC 5103 Bidirectional Flow Export Using IP Flow Information Export (IPFIX) секция 2 Terminology // URL: https://tools.ietf.org/html/rfc5103 (дата обращения: 13.05.2020)

19. NetFlow - Wikipedia раздел 1.1 “Network Flows” //URL: https://en.wikipedia.org/wiki/(дата обращения: 13.05.2020)

20. RFC 3954 Cisco Systems NetFlow Services Export Version 9 секция 3.1. “The NetFlow Process on the Exporter” // URL: https://tools.ietf.org/html/rfc3954 (дата обращения: 13.05.2020)

21. RFC 3954 Cisco Systems NetFlow Services Export Version 9 секция 3.2. “Flow Expiration” // URL: https://tools.ietf.org/html/rfc3954 (дата обращения: 13.05.2020)

22. ahlashkari/ISCXFlowMeter // URL: https://github.com/ahlashkari/ISCXFlowMeter/ (дата обращения: 13.05.2020)

23. ISCXFlowMeter/FlowGenerator.java at master · ahlashkari/ISCXFlowMeter · GitHub // URL: https://github.com/ahlashkari/ISCXFlowMeter/blob/master/src/main/java/iscx/cs/unb/ca/ifm/flowgen/FlowGenerator.java (дата обращения: 13.05.2020)

24. «Программа классификации компьютерных атак по набору данных ISCX Botnet». Техническое задание.

25. «Программа классификации компьютерных атак по набору данных ISCX Botnet». Руководство оператора.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Изм.	Номера листов (страниц)				Всего листов (страниц) в документе	№ документа	Входящий № сопроводительного документа и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					